

# Signaling to Establish the Connection between Diverse Devices in WLAN

J. T. Anita Rose  
Research Scholar  
Manonmaniam Sundaranar  
University, TN, India.

A. Chandrasekar, PhD  
Professor  
St. Joseph's College of  
Engineering, TN, India.

D. Frederick Swartz  
Research Scholar  
Anna University,  
TN, India.

## ABSTRACT

As the IMS provides session establishment, QoS parameter negotiation, authentication and accounting it is an overlay architecture well qualified for session signaling concerning video in everywhere scenarios. The management of the signaling can be integrated as an application into a home media gateway (HMG) server. Heterogeneous mobile devices working with the diverge of codec leads to signaling disconnection because of parameter mismatch. The HMG server is further responsible for all necessary media signaling towards a proper media server to support the interoperation so that it can identify the disconnection because of SDP parameter mismatch further modify the SDP parameter to establish the connection. HMG also convert the media payload from one format to another so that user equipment able play the video streams. The media signaling requires a dedicated session description and can be provided by SIP or RTSP. This paper proposes a general IMS-based framework for delivering video in everywhere services over several access networks and presents the corresponding signaling flows for on-demand as well as live video calls. It further presents an approach for using SIP and RTSP in cooperation for realizing video on demand scenarios.

## Keywords

Cooperative Media Streaming, On-Demand Signaling, Live Video Signaling, Connection Establishment for Transcoding.

## 1. INTRODUCTION

As IMS (IP Multimedia System) are more and more improving in terms of bandwidth and quality of service (QoS) and is the right environment in which to deliver the video in everywhere (ViE) service. Apart from granting users the ability to access their services using different devices and access technologies, the major goal of IMS is the delivery of multimedia services everywhere. There are different proposals of implementation standards for IMS by different standard bodies also different specifications for the delivery of multimedia, again from different bodies. However, the specification that has received the greatest interest from the research community for the development and testing of multimedia services is the one proposed by ETSI TISPAN [1]. Multimedia session delivery involves the use of a session control protocol to control the session and a media control protocol to control the media delivery.

The media delivery in a standard IMS architecture, for example, is carried out by what is known as the Media Resource Function (MRF), consisting of two distinct parts, namely the MCF (Media Controller Function) and MDF (Media Delivery Function) [2]. The ViE specification also has a similar component for media control, which is to be implemented by a Media Control (MC) server and the media

delivery unit to be implemented by a fully-fledged Media Server (MS). In addition to the media delivery and control units, IMS services are controlled by a service controller unit is called Service Control Function (SCF). Basically, this unit is a SIP application server (AS). So, if a user knows the service description of a given ViE service, s/he contacts the SCF to obtain the desired service. The SCF, in turn, will contact the MC server to initiate the delivery of media. The MC server then initiates the media delivery by instructing the MS to send the requested stream directly to the user. In general, the MC server initiates the media request on behalf of the user and the media server, MS delivers the stream to the user (not to the initiator of the session). As all media requests pass through the MC server, this means that if one were to follow the specification directly, all media requests including session initiation should pass through MC server is shown in figure 1.

User device application might be implemented using web real time communication (WebRTC) [3]. WebRTC is an open source framework can be easily integrated in end user application which might be either based on browser or based on native device specific. WebRTC integrated with the application level Session Initiation Protocol (SIP) to establish the connection between the devices by negotiation [4]. SIP Session negotiation can occur using Session Description Protocol (SDP) to find out if the other peer is reachable and if it is willing to establish the connection.

In this paper, we explained the connection establishment components for the streaming of video in on-demand scenario as well as live-video calls scenario. In SIP signaling, the user agent client (UAC) and user agent server (UAS) must exchange the SDP and negotiate for the connection establishment. Heterogeneous mobile devices working with the diverge of codec leads disconnection because of mismatch in SDP. To overcome the disconnection requires a dedicated media conversion management which acts as transcoding instance between UAC and UAS. Such a transcoding instance must be hosted in home media gateway (HMG) server. HMG act as a Back to Back User Agent (B2BUA) to do the interoperation support such as: First, Identify the disconnection because of SDP parameter mismatch and modify the SDP parameter to establish the connection. Second, convert the media payload from one format to another so that UE could play the streams.

The remainder of this paper is organized as follows. Section II reviews of related work. Section III develops the SIP signaling sequence for on-demand and live video streaming. Section IV gives the implementation and discussion. Section V concluding remarks.

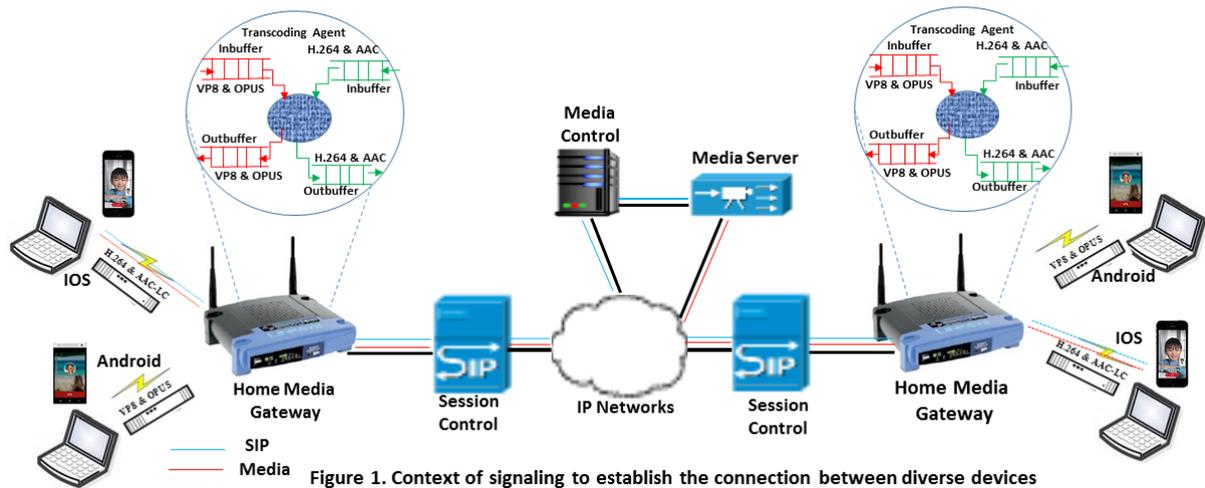


Figure 1. Context of signaling to establish the connection between diverse devices

## 2. RELATED WORK

The authors in [5] showed how a new SIP header (called SIP-MEX) and new SIP bodies (an XML document in the SIP INFO message) can be used to send media control commands to the MCF. On the other hand, other researchers have also suggested the integration of SIP and RTSP to create a comprehensive media control protocol [6]. However, as mentioned in [7], to avoid the IMS signaling procedures causing extra delays, it is always necessary to define a clear separation between service/session control performed at IMS level and media flow control handled end-to-end between user equipment and the content service. This could be one of the reasons that ETSI-TISPAN proposed a different media control protocol other than SIP in the standard specification.

In general, those who have proposed SIP as a media control protocol have tried to justify their proposal from the point of view of media control requirements that cannot be handled by RTSP and also for handling bandwidth reservation requests and responses. Nevertheless, as to the support of bandwidth negotiation, the IETF has developed extensions to SDP [8] and it should no longer be a problem to use RTSP. However, the advantage of using RTSP as a media control protocol is that the MCF is not required to translate media control commands received from the UE when it forwards them to the MDF. The concept of an RTSP gateway is also presented in [9], where the authors proposed a gateway that converts RTSP messages to SIP messages and vice versa. On the other hand, the use of an RTSP proxy for the delivery of streaming service for UEs without RTP support is presented in [10].

SIP is an application level signaling protocol, widely used for VoIP and videoconferencing over IP networks [11]. Before establishing a Peer-to-Peer Connection connectivity checks or session negotiation can occur, we must find out if the other peer is reachable and if it is willing to establish the connection. We must extend an offer, and the peer must return

an answer. The signaling server can act as a gateway to an existing communications network, in which case it is the responsibility of the network to notify the target peer of a connection offer and then route the answer back to the client initiating the exchange. if both peers are connected to the same signaling service, then the service can shuttle messages between them. Client application uses Session Description Protocol (SDP) to establish a peer-to-peer connection, both peers must exchange of SDP descriptions to their respective audio, video, and other data [12]. SDP does not deliver any media itself; instead it is used to describe the "session profile,"

which represents a list of properties of the connection: types of media to be exchanged (audio, video, and application data), network transports, used codecs and their settings, bandwidth information, and other metadata.

WebRTC specification requires that all transferred data-audio, video, and custom application payloads must be encrypted while in transit [3]. The combination of the record sequence number, offset, and retransmission timer allows DTLS to perform the handshake over UDP. To complete this sequence, both network peers generate self-signed certificates and then follow the regular TLS handshake protocol. WebRTC provides media acquisition and delivery as a fully managed service: from camera to the network, and from network to the screen. The WebRTC application specifies the media constraints to acquire the streams and then registers them with the Peer Connection object. From there, the rest is handled by the WebRTC media and network engines provided by the browser: encoding optimization, dealing with packet loss, network jitter, error recovery, flow, control, and more.

we hope that our proposal will be helpful to researchers wishing to develop a ViE service using streaming servers particularly as it conforms to the specification. In fact, the home media gateway also implements new functionalities such as transcoding media for on-demand and live video calls services for use by service developers.

## 3. SIGNALING TO ESTABLISH A CONNECTION

The logical network end-point such as mobile phone, lab top and tablet PC connected in local network are behind a NAT. As the addresses and ports are private can't establishes a SIP based session across the Internet. Session Traversal Utilities for NAT (STUN) is a standardized set of methods and a network protocol to allow an end-point to discover its own public IP address and the port from a STUN server on the internet. SIP works in conjunction with several other application layer protocols that identify and carry the session media. Media identification and negotiation is achieved with the Session Description Protocol (SDP). For the transmission of media streams (voice, video) SIP typically employs the Real-time Transport Protocol (RTP).

SIP is a signaling protocol widely used in VoIP to control and manage calls. A typical SIP environment consists of three basic components User Agent Client (UAC), User Agent Server (UAS) and Back to Back User Agent (B2BUA). The B2BUA act as a SIP Proxy Server. A SIP user agent (UA) is a

logical network end-point used to create or receive SIP messages and there by manage a SIP session. A SIP UA can perform the role of a UAC which initiate and sends the SIP requests to the UAS which receive and process the requests then returns a SIP response. The B2BUA is an intermediary entity that acts as both a server and a client for making requests on behalf of other clients. A B2BUA primarily plays the role of routing, meaning that its job is to ensure that a

request is sent to another entity closer to the targeted user. B2BUA are also useful for enforcing policy, such as for determining whether a user can make a call. A B2BUA interprets, and, if necessary, rewrites specific parts of a request message before forwarding it. In our system, we implement the B2BUA module on home media gateway (HMG) server to rewrites the codec specific parts of a request message before forwarding to UAC.

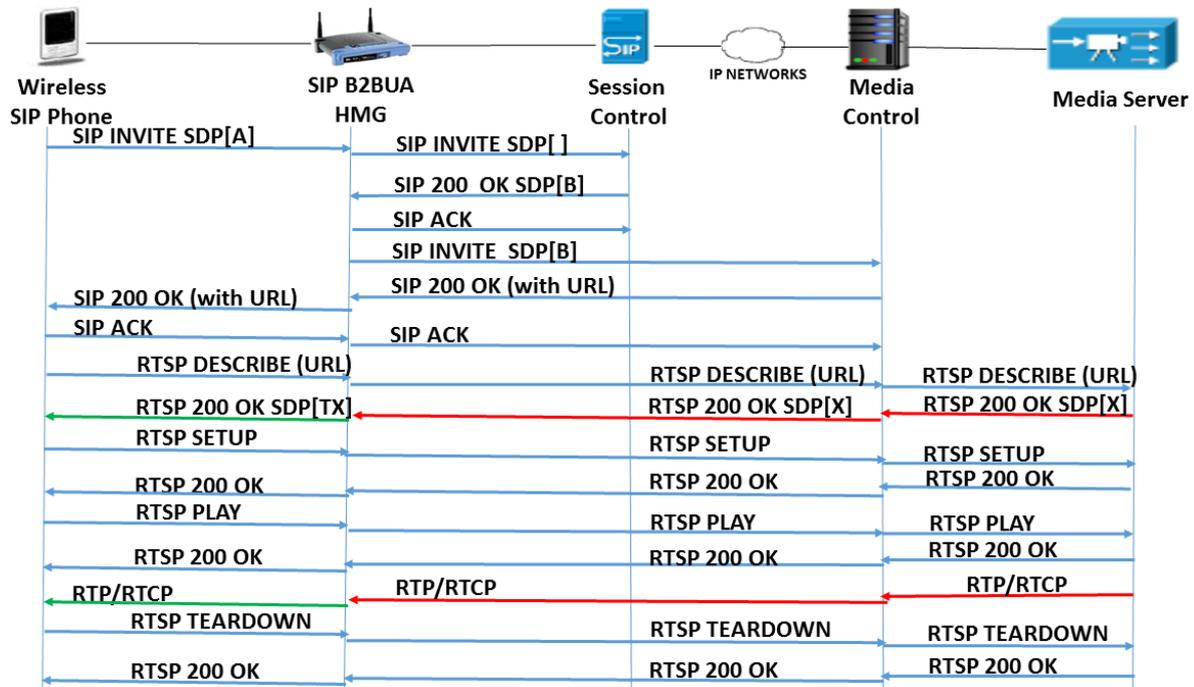


Figure 2. SIP connection Establishment Signals for on-demand Streaming

### 3.1 Signaling for On-Demand Video Streaming

In on-demand video delivery the user equipment (UE) such as mobile device must select the URL of the media from the URL list published by the server. The Session Control (SC) server prepare the SDP with the URL of the Media Control (MC) and send the replay to HMG. The HMG forward the SIP INVITE to MC server, The MC upon receipt of an INVITE from HMG, it includes the URL list of the media in the SIP OK message then send to the HMG. The HMG forwards the SIP Ok message to the UE. The user can have the option to select the URL and the UE sends the RTSP DESCRIBE commands with the selected URL to the HMG. The HMG forwards the DESCRIBE commands to MS through the MC. The MS prepare the SDP and send the RTSP OK message with SDP to HMG through the MC. The HMG evaluate the received SDP and decide whether the transcoding is required. The transcoding is required when the UE do not have the capability to play the media with received SDP so that the HMG the prepare the SDP to enable transcode.

The RTSP SETUP commands to the media server to initiate and set up the media session. The UE send the SETUP commands to the MC through the HMG. When the MC received the SETUP command from the UE, it forwards the request to the media server. The MC uses the destination parameter of the RTSP protocol, to pass on the destination of the media, i.e., the address of the UE. This parameter is included in the SETUP command of the request. The MC obtains client address information from the SDP payload of

the SIP INVITE command. Accordingly, the MC forwards the request to the media server and upon receipt of the media, delivers it to the UE through HMG. Session handling is one important aspect of media servers. An RTSP session initiation request (ex, DESCRIBE) may not necessarily end up in an RTSP session. The client may not be able to play the media (video) if it does not support the codec that the media is encoded in. As a result, even though there is an I/O (network) session between client and server, an RTSP session is basically created when the client sends a SETUP request to the server. This tells the server that the client can play the media and the server generates and sends a unique session ID within the response is shown in figure 2.

The RTSP PLAY commands to the media server to start the media streaming. The UE send the PLAY commands to the MC through HMG. When the MC received the PLAY command from the UE, it forwards the request to the media server. Media server start the streaming with the URL specified during the session initiation request (ex, DESCRIBE). The RTSP TEARDOWN commands to the media server to STOP the media streaming.

### 3.2 Signaling for Live Video Streaming

In live video chat, before establishing a peer-to-peer connection between UE A and UA B, connectivity checks or session negotiation can occur, the peer must find out if the other peer is reachable and if it is willing to establish the connection. peer must extend an offer and return an answer. The B2BUA is the responsibility of the network to notify the target peer of a connection offer and then route the answer

back to the client initiating the exchange. B2BUA uses SDP messages to pass the parameter and establish a peer-to-peer connection, both peers must exchange of SDP descriptions to their respective audio, video, and other data streams. B2BUA allows SIP messages from the UAC of originating party to be terminated at the one side of the network, where the SDP media parameter is verified in the message. Any mismatch in the received SDP message then the message is recomposed to include the information needed for the SIP to establish a VoIP call shown in figure 3.

The caller initiate the video call by send the SIP INVITE with its expected SDP[A] to the callee devices thru HMG. In HMG, does not forward the received caller SDP[A] until receive the callee devices expected SDP[B] to overcome the call disconnection because of SDP mismatch. The HMG send

the SIP INVITE without SDP [ ] to the callee device. Upon receive the INVITE without SDP [ ] the callee device does all the usual stuff to process the INVITE by sending a SIP 180 ringing message, but is unaware of what codec might be involved in the session. The callee answered by sending a SIP 200 Ok with expected SDP [B]. HMG knows the expected SDP of both devices after receive the expected SDP [B] of the callee devices. At this stage, HMG validate both the SDP and find out if there is any codec mismatch in caller SDP[A] and callee SDP[B]. HMG enable the transcoding if found the codec mismatch in SDP, that is to send the SIP 200 Ok with SDP[TA] to the caller and send the SIP ACK with SDP[TB] to the callee. With this changes caller and callee could establish the media channel to transmit the RTP media messages.

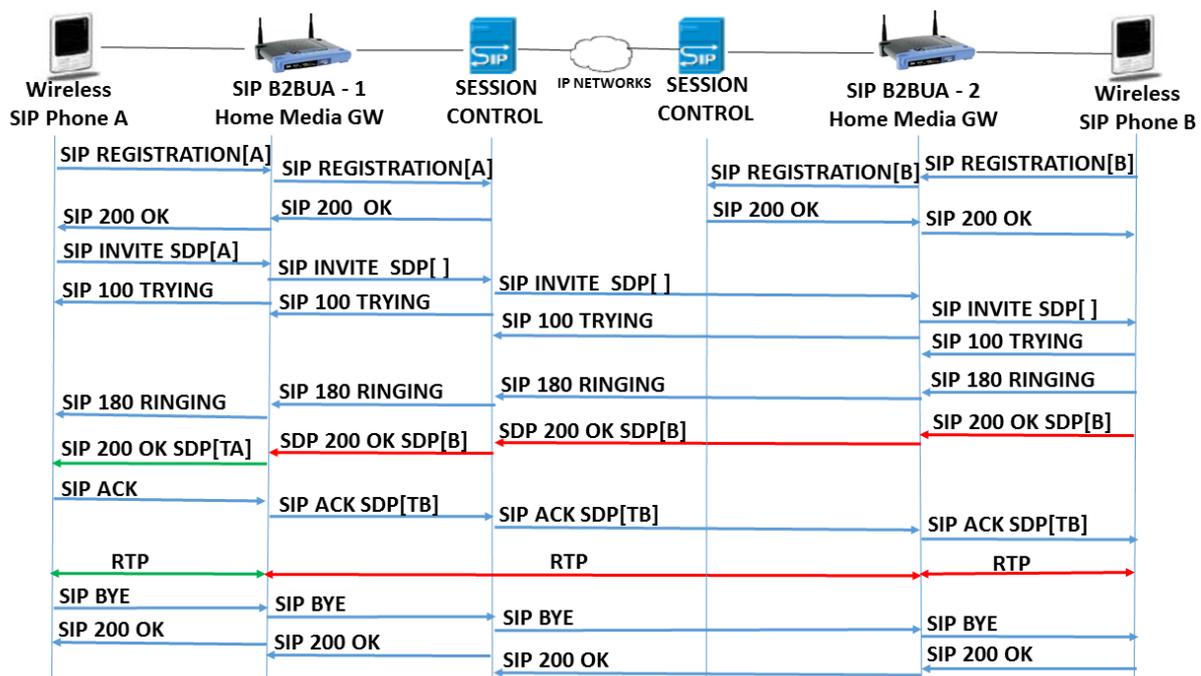


Figure 3. SIP connection Establishment Signals for Live Video Calls

The HMG will set the global flag doTrans as true when the media channel has established between UE A and UE B by activate the transcode. The media is transmitted through RTP channel between the UE A & UE B and is carried by the RTP. Finally, when either of the UE wants to end the call, it should send a SIP BYE message to the other UE through the HMG. The B2BUA of the HMG release the resources occupied during the call.

#### 4. DISCUSSION

To conduct the experiments, a smartphone with 1.2 Ghz CPU and 1 GBRAM running Android 4.0.3 version was used as the end user device. Four desktop PC, with 3.1 GHz quad-core CPU and 8 GB RAM each, represented the session border controller (SBC) server, home media gateway (HMG), media control (MC) and media server (MS). All the desktops used linux OS, HMG ran exGate application with transcoding agent, SIP B2BUA and RTSP proxy server. SBC ran the exGate application with SIP application server. MS ran the streaming server application. The time was recorded when the HMG received a request and when the same request left the HMG to the streaming server. The difference was calculated to determine the delay through the HMG. The same approach was used for the responses. The average delay introduced by

the HMG was found to be negligible (close to 40 Nano seconds). Thus, we believe that the HMG can be considered a good solution for those who wish to use a streaming server for media delivery in IMS based ViE services, as it does not introduce much delay into the whole system.

As mentioned previously, among the many advantages of having a separate RTSP proxy in place is the ability to change media within an existing session (e.g., Transcoding), which can be done easily without the involvement of the UE. This is important functionality because the UE does not require a different connection to obtain the new stream as the HMG can handle that itself. The AS can initiate the modification of media within an existing session based on different sets of rules and the RTSP proxy can deliver the modified stream without involving the UE. We believe this will help researchers to implement new and innovative services that can be implemented by any standard UE. Because of the nature of the RTSP protocol, if the media source disappears for whatever reason, there is no way that the MCF can know about the situation. On the other hand, if we use a RTSP proxy, because the HMG also handles RTP packets, it knows if the session is alive or dead and can take the appropriate action immediately when a problem arises.

## 5. CONCLUSION

The media streaming research community mainly uses streaming servers for the delivery of media for ViE services. On the other hand, the IMS-based ViE specification specifies that a streaming session is initiated by the MCF on behalf of the user. Nevertheless, most open source streaming servers do not allow the initiation of a streaming session by a different client to the RTSP client intending to consume the stream. To overcome this limitation, a HMG proxy as described in this paper, can be used as a work around thereby enabling researchers to use the available streaming servers while adhering to the standard. The RTSP proxy can be integrated into the MCF or be deployed as a separate entity. According to timing experiments conducted, the proxy does not introduce a significant delay to the service delivery process and as such the authors believe it to be a good solution. In addition to allowing the use of streaming servers as MDF units for the delivery of ViE services, the paper also presented some of the additional benefits arising from use of the proxy.

## 6. REFERENCES

- [1] ETSI TS 182 027: IPTV Architecture; IPTV functions supported by the IMS subsystem, March 2011.
- [2] R. Spiers, R. Marston, R Good and N. Ventura (2009). The UCT IMS IPTV Initiative. In Proceedings of the 2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies. pp. 503-508. 2009.
- [3] C. Holmberg, S. Hakansson, G. Eriksson, "Web Real-Time Communication Use Cases and Requirements", Internet Engineering Task Force, RFC: 7478, March 2015
- [4] M. Handley, V. Jacobson, "SDP: Session Description Protocol," Network Working Group, RFC: 2327, April 1998.
- [5] S. Sivasothy, G. M. Myoung, and N. Crespi (2009). A unified session control protocol for IPTV services. In *Proceedings of the 11th International Conference on Advanced Communication Technology*. pp. 961-965, February 15-18, 2009, Gangwon-Do, South Korea.
- [6] R. G. Shiroor (2007). IPTV and VoD services in the context of IMS. In *International Conference on IP Multimedia Subsystem Architecture and Applications*, pp. 1-5, December 6-8, 2007.
- [7] B. Chatras, M. Saïd. Delivering Quadruple Play with IPTV over IMS. [www.icin.biz/files/programmes/Session8A-1.pdf](http://www.icin.biz/files/programmes/Session8A-1.pdf) Accessed: July 1, 2010. [3] J. Tang and Y. Cheng, "Quick detection of stealthy SIP flooding attacks in VoIP networks," in *Proc. IEEE ICC*, Kyoto, Japan, Jun. 5–9, 2011, pp. 1–5.
- [8] Internet draft. SDP media capabilities Negotiation draft-ietf-mmusic-sdp-media-capabilities-09. <http://tools.ietf.org/html/draft-ietf-mmusic-sdpmedia-capabilities-09>. Expires: August 2010.
- [9] C. Riede, A. Al-Hezmi and T. Magedanz (2008). Session and media signaling for IPTV via IMS. In *Proceedings of the 1st International Conference on Mobile Wireless Middleware, Operating Systems, and Applications*. February 13 - 15, 2008. Brussels, Belgium.
- [10] Z. Shibeshi, A. Terzoli, and K. Bradshaw (2010). Streaming Session Transfer between Registered User Agents. In *SATNAC'10: Proceedings of the 13th Southern African Telecommunications Networks and Applications Conference*. Spier Estate, Stellenbosch, South Africa, 5 to 8 September 2010.
- [11] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol" Network Working Group, RFC: 3261, June 2002.
- [12] M. Handley, V. Jacobson, "SDP: Session Description Protocol," Network Working Group, RFC: 2327, April 1998.

## 7. AUTHOR PROFILE

**Anita Rose J. T.**, received her Master's degree in Computer Science and Engineering from Annamalai University, India in 2005 and pursuing Ph.D in Computer science & Engineering in Manonmaniam Sundaranar University, India. Currently working as an Associate Professor in the Department of Computer Science & Engineering, St. Joseph's College of Engineering, Chennai. Her area of interest includes Cooperative computing, Energy efficient computing.

**Frederick Swartz Daniel D.**, received his Master's degree in Computer Science and Engineering from Anna University, India in 2005 and pursuing Ph.D in Information and Communication Engineering in Anna University, Chennai. He worked in software companies Tech Mahindra R&D Services Ltd, Larsen & Tubro Technologies Ltd, etc. Currently working as a Lead Engineer in Mundio Mobiles. His area of interest includes Parallel and Distributed Programming, Cooperative Multimedia Streaming and Internet of Things in Embedded System.

**Dr. A. Chandra Sekar**, received his Master's degree in Computer Science and Engineering from Madurai Kamaraj University, India and Ph.D in Information and Communication Engineering from Anna University, India. He is currently working as a professor in the Department of Computer Science & Engineering in St. Joseph's College of Engineering, Chennai, India. His area of interest includes Network Security, Web Services and Analysis of Algorithm.